

Hacking Articles

Raj Chandel's Blog

[Menu](#)

[Home](#) » [Database Hacking](#) » [Detect SQL Injection Attack using Snort IDS](#)

Database Hacking , Penetration Testing

Detect SQL Injection Attack using Snort IDS

January 11, 2018 By Raj Chandel

Hello friends!! Today we are going to discuss how to “Detect SQL injection attack” using Snort but before moving ahead kindly read our previous both articles related to Snort Installation (**Manually** or using **apt-respiratory**) and its **rule configuration** to enable it as IDS for your network.

Basically In this tutorial we are using snort to capture the network traffic which would analysis the SQL Injection quotes when injected in any web page to obtain information of database system of any web server. Snort will generate the alert for malicious traffic when caught those traffic in its network and network administrators will immediately get attentive against suspicious traffic and could take effective action against the attacking IP.

Requirement

IDS: Snort (Ubuntu)

Web application: Dhakkan

You can configure your own web server by taking help of our article “Configure Web server for penetration testing”

Let's Begin!!

Identify Error Based SQL Injection

As we know in Error based SQL injections the attacker use **single quotes** (') or **double quotes** (") to break down SQL query for identify its vulnerability. Therefore be smart and add a rule in snort which will analyst Error based SQL injection on the server when someone try to execute SQL query in your network for unprivileged access of database.

Execute given below command in ubuntu's terminal to open snort local rule file in text editor.

```
sudo gedit /etc/snort/rules/local.rules
```

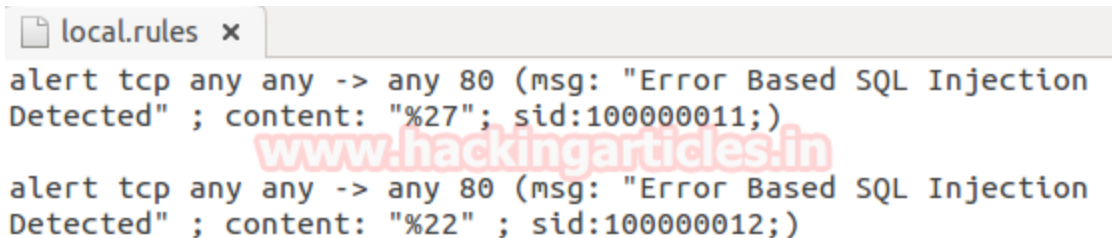
Now add given below line which will capture the incoming traffic coming on any network IP via port 80.

```
alert tcp any any -> any 80 (msg: "Error Based SQL Injection Detected"; content:  
alert tcp any any -> any 80 (msg: "Error Based SQL Injection Detected"; content:
```

If you read above rule you can notice that I had applied filter for content "%27" and %22 are URL encoded format use in browser for single quotes(') and double quotes (") respectively at the time of execution of URL.

Turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



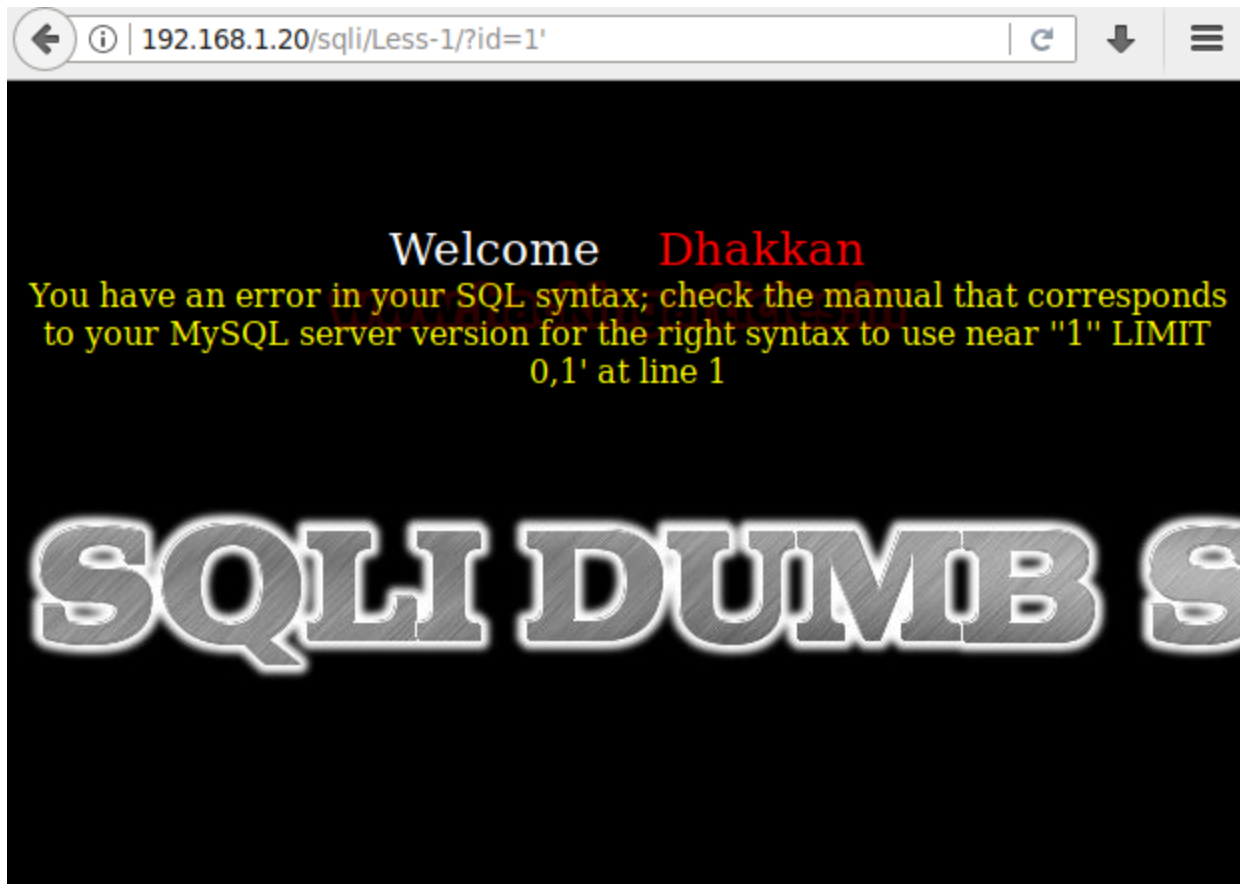
```
local.rules x  
alert tcp any any -> any 80 (msg: "Error Based SQL Injection  
Detected" ; content: "%27"; sid:100000011;)  
alert tcp any any -> any 80 (msg: "Error Based SQL Injection  
Detected" ; content: "%22" ; sid:100000012;)
```

Now test your above rule by making Error based sql injection attack on web application "Dhakkan", therefore open the server IP in web browser and use single quotes (') for identify SQL injection vulnerability as shown below.

```
192.168.1.20/sqli/Less-1/?id=1'
```

For more detail on Error Based SQL injection read our previous article.

Now when attacker will execute malicious quotes in browser for testing Error Base SQL injection then the IDS of the network should also capture this content and will generate the alert.



As per our prediction from given image you can observe the snort has generated alert for Error Based sql injection when capture malicious quotes.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

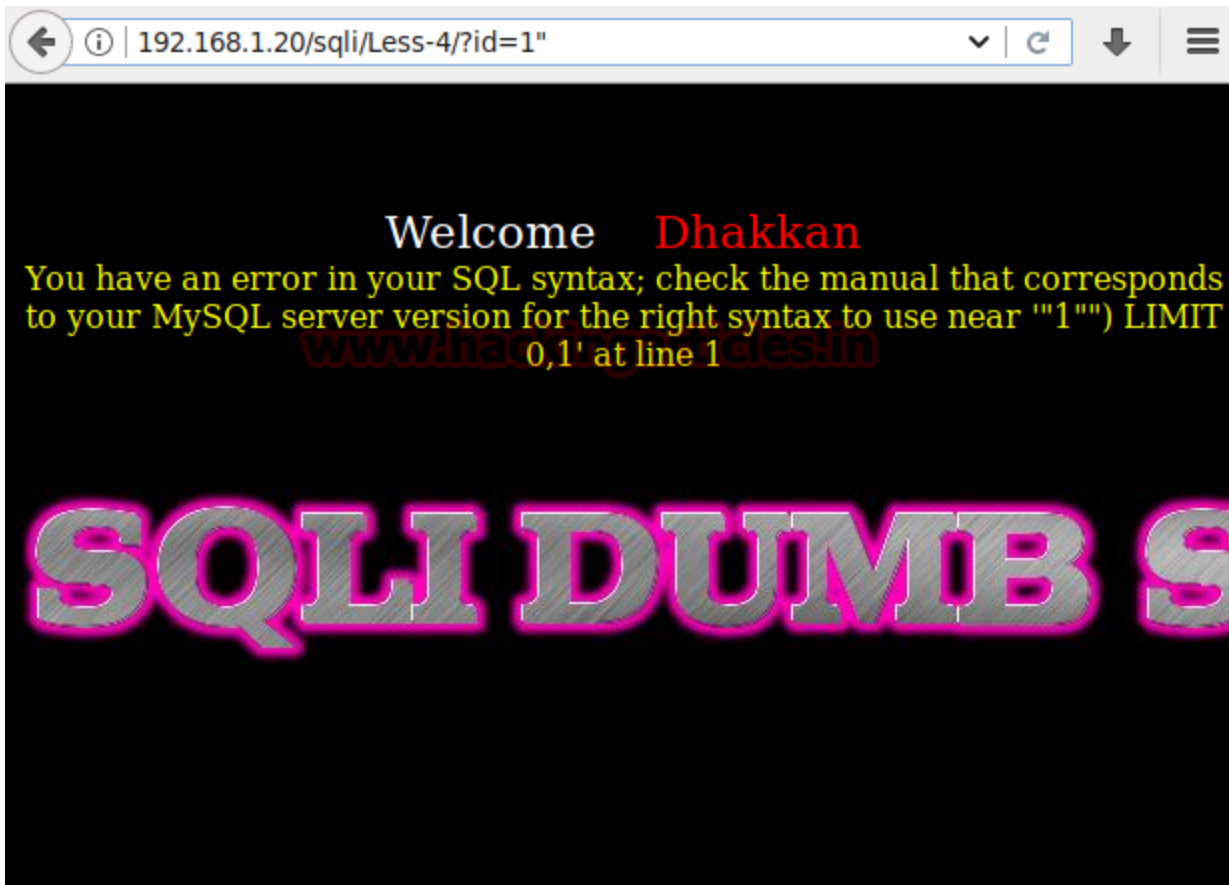
```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-03:21:25.445355  [**] [1:100000011:0] Error Based SQL Injection Detected [
**] [Priority: 0] {TCP} 192.168.1.21:37486 -> 192.168.1.20:80
```

Testing Double Quotes Injection

Now again open the server IP in web browser and use double quotes (") for identify SQL injection vulnerability as shown below.

```
192.168.1.20/sqli/Less-4/?id=1"
```

Now when attacker will execute malicious quotes in browser for testing Double quotes SQL injection then the IDS of the network should also capture this content and will generate the alert.



From given image you can observe the snort has generated alert for Error Based sql injection when capture malicious quotes.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming from 192.168.1.21 on port 80.

```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-03:22:14.485463  [**] [1:100000012:0] Error Based SQL Injection Detected [
**] [Priority: 0] {TCP} 192.168.1.21:37488 -> 192.168.1.20:80
```

Boolean Based SQL Injection

As we know in Boolean based SQL injections the attacker use **AND /OR** operators where attacker will try to confirm if the database is vulnerable to Boolean SQL Injection by evaluating the results of various queries which return either TRUE or FALSE.

Now add a rule in snort which will analyse Boolean based SQL injection on the server when someone try to execute SQL query in your network for unprivileged access of database. Here I **had applied filter for content "and" & "or" to be captured. Here nocase denotes not case sensitive it can be as AND/and, OR/or.**

```
alert tcp any any -> any 80 (msg: "AND SQL Injection Detected"; content: "and"  
alert tcp any any -> any 80 (msg: "OR SQL Injection Detected"; content: "or" ;
```

Turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

local.rules x

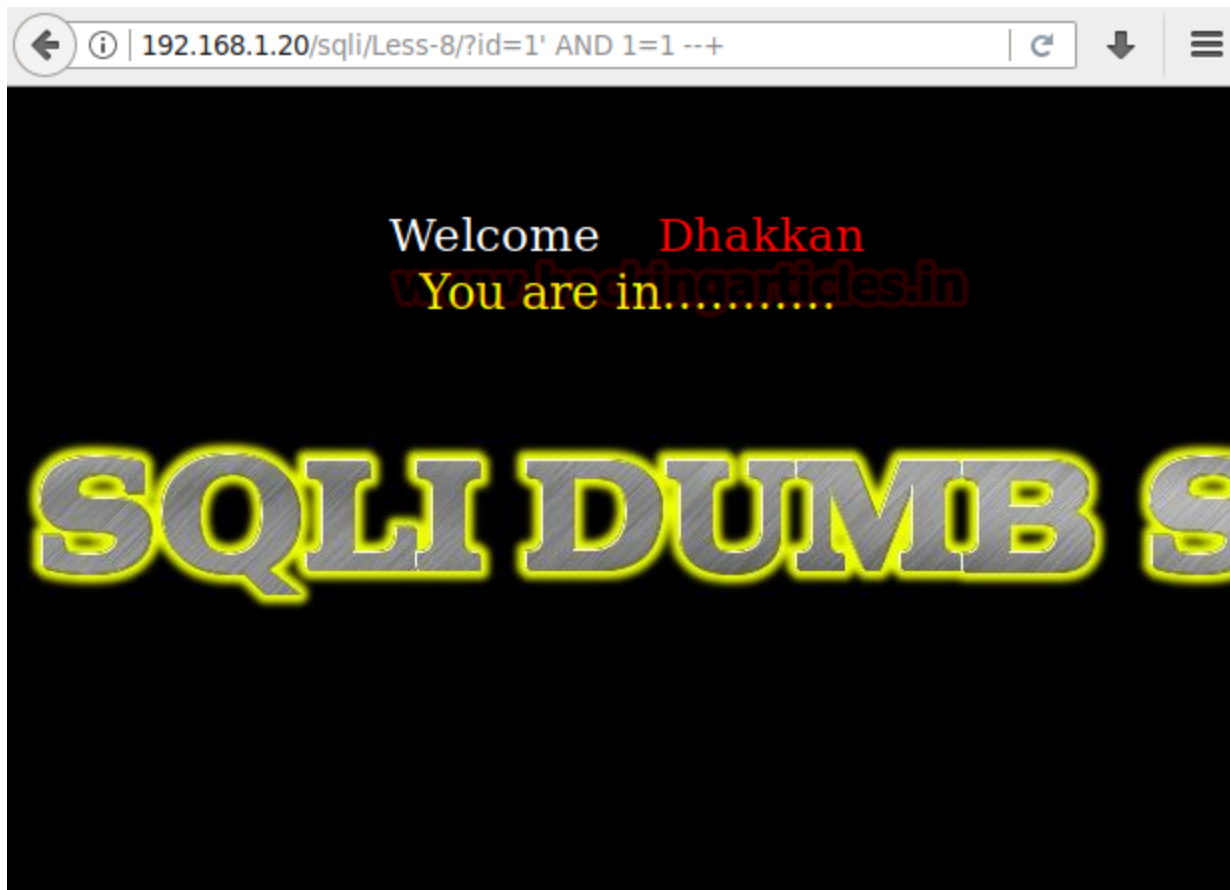
```
alert tcp any any -> any 80 (msg: "AND SQL Injection Detected" ;  
content: "and"; nocase ; sid:100000060;)  
alert tcp any any -> any 80 (msg: "OR SQL Injection Detected" ;  
content: "or"; nocase ; sid:100000061;)
```

Again open the server IP in web browser and use AND operator for identify Boolean SQL injection vulnerability as shown below.

```
192.168.1.20/sqli/Less-8/?id=1' AND 1=1 --+
```

For more detail on Boolean Based SQL injection read our previous article.

Now when attacker will execute malicious quotes in browser for testing Boolean Base SQL injection then the IDS of the network should also capture this content and will generate the alert.



Testing OR Operator

As per our calculation from given image you can observe the snort has generated alert for Boolean Based sql injection when captured content AND.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-03:25:02.114714  [**] [1:100000060:0] AND SQL Injection Detected [**] [Priority: 0] {TCP} 192.168.1.21:37492 -> 192.168.1.20:80
```

Again open the server IP in web browser and use OR operator to identify Boolean SQL injection vulnerability as shown below.

```
192.168.1.20/sqli/Less-8/?id=1' OR 1=1 --+
```

Now when attacker will execute malicious quotes in browser for testing Boolean Base SQL injection then the IDS of the network should also capture this content and will generate the alert.



As per our calculation from given image you can observe the snort has generated alert for Boolean Based sql injection when captured content OR.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-03:11:36.167866  [**] [1:100000061:0] OR SQL Injection Detected [**] [Priority: 0] {TCP} 192.168.1.21:37474 -> 192.168.1.20:80
```

Encoded AND/OR

Similarly in given below rule I had applied filter for content “%26%26” and “%7c%7c” are URL encoded format use in browser for && and || respectively at the time of execution of URL.

```
alert tcp any any -> any 80 (msg: "AND SQL Injection Detected"; content: "and" ;
alert tcp any any -> any 80 (msg: "OR SQL Injection Detected"; content: "or" ;
```

Turn on IDS mode of snort by executing given below command in terminal:


```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

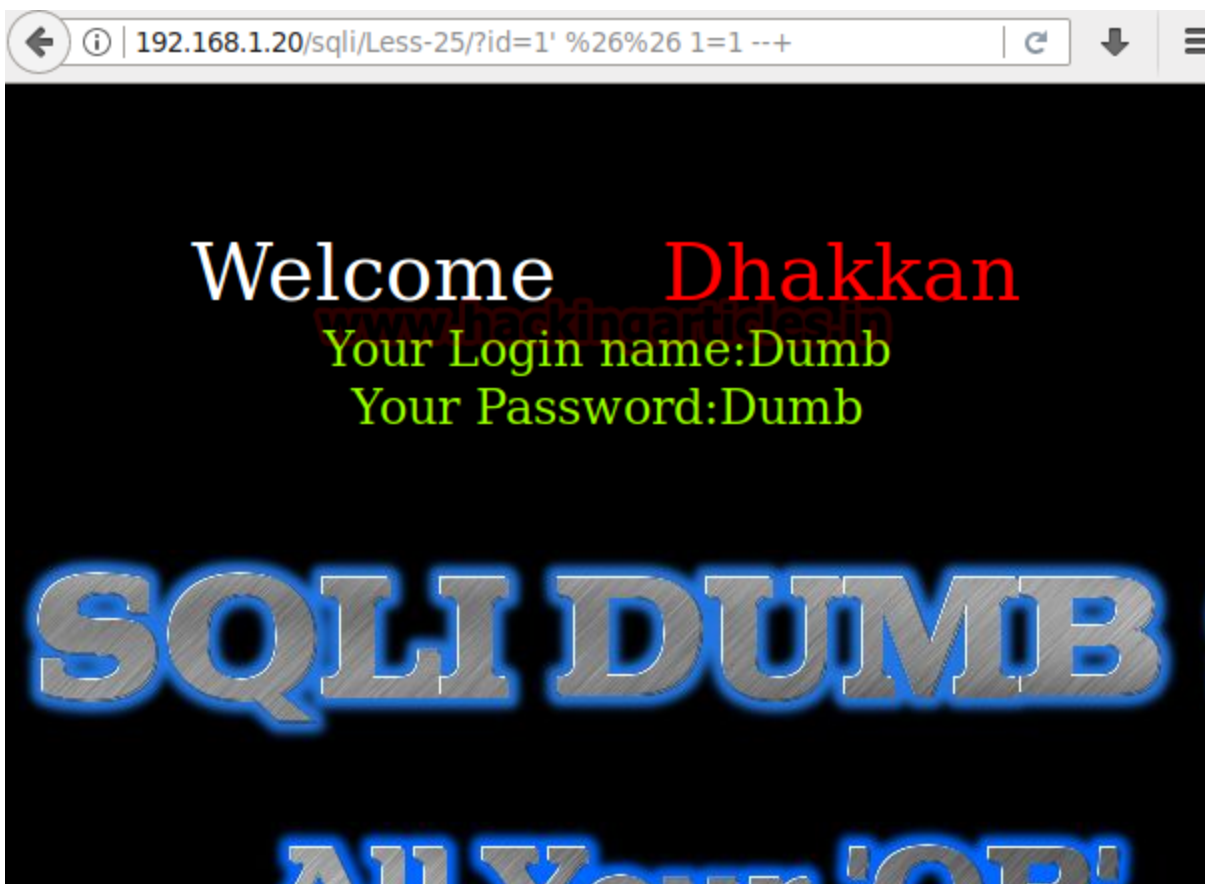
```
local.rules x
alert tcp any any -> any 80 (msg: "AND SQL Injection Detected" ;
content: "%26%26"; sid:10000008;)
alert tcp any any -> any 80 (msg: "OR SQL Injection Detected" ;
content: "%7C%7C"; sid:10000009;)
```

Again open the server IP in web browser and use **&& operator** for identify Boolean SQL injection vulnerability as shown below.

```
192.168.1.20/sqli/Less-25/?id=1' %26%26 1==1 --+
```

For more details read our previous article

Now when attacker will execute malicious quotes in browser for testing Boolean Base SQL injection then the IDS of the network should also capture this content and will generate the alert.



As per our calculation from given image you can observe the snort has generated alert for Boolean Based sql injection when captured content **%26%26**.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

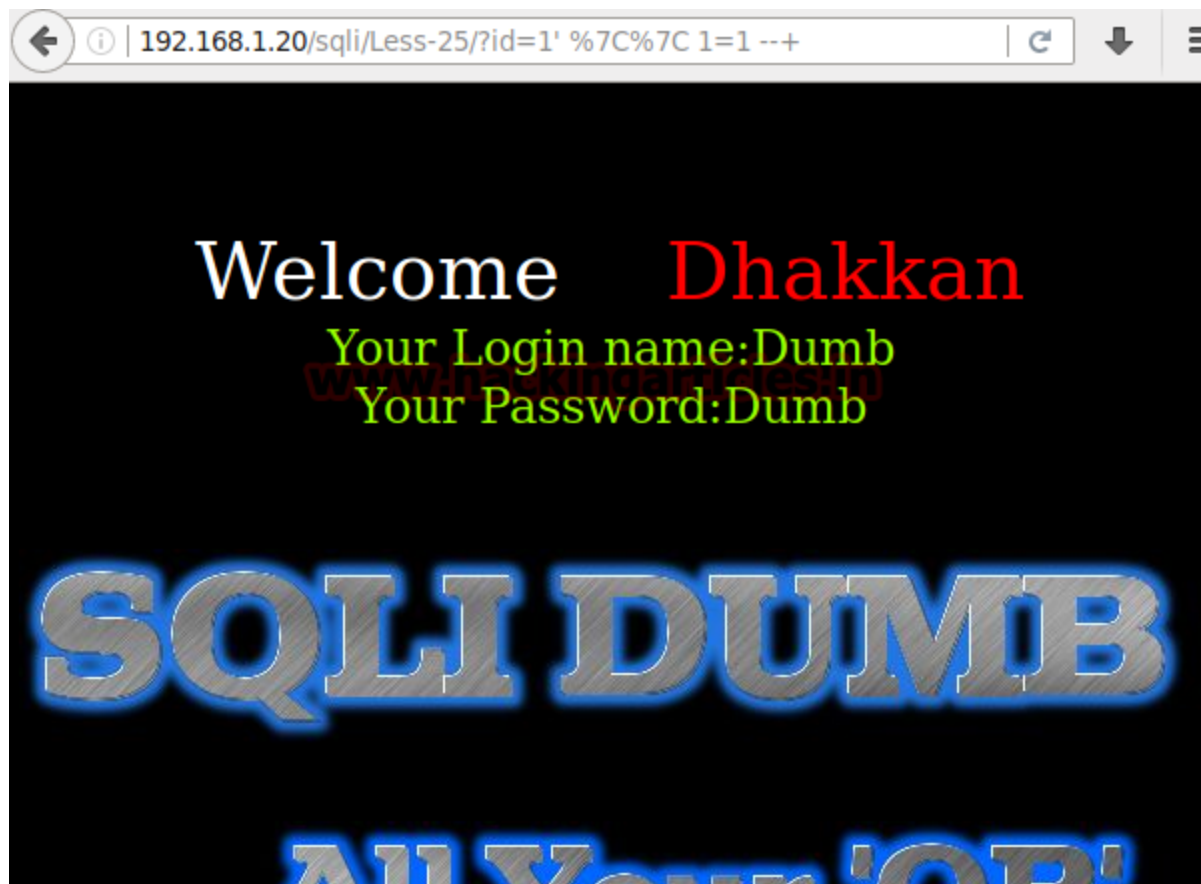
```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-03:00:28.491295  [**] [1:10000008:0] AND SQL Injection Detected [**] [Priority: 0] {TCP} 192.168.1.21:37454 -> 192.168.1.20:80
```

Testing Encoded OR Operator

Again open the server IP in web browser and use **|| operator** for identify Boolean SQL injection vulnerability as shown below.

```
192.168.1.20/sqli/Less-25/?id=1' %7C%7C 1==1 --+
```

Now when attacker will execute malicious quotes in browser for testing Boolean Base SQL injection then the IDS of the network should also capture this content and will generate the alert.



As per our calculation from given image you can observe the snort has generated alert for Boolean Based sql injection when captured content **%7C %7C**.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-03:01:34.669653  [**] [1:10000009:0] OR SQL Injection Detected [**] [Priority: 0] {TCP} 192.168.1.21:37456 -> 192.168.1.20:80
```

Identify Form Based SQL Injection

The Form Based SQL injection also known as “Post Error based SQL injection” because the attacker executes malicious quotes inside Login form of a web page that contains text field for username and password to login inside web server.

Therefore now add a rule in snort which will analyst Form based SQL injection on the server when someone try to execute SQL query in your network for unprivileged access of database.

```
alert tcp any any -> any 80 (msg: "Form Based SQL Injection Detected"; content
```

If you read above rule you can notice that I had applied filter for content “%27” to be captured; turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```

```
local.rules x
alert tcp any any -> any 80 (msg: "Form Based SQL Injection
Detected" ; content: "%27" ; sid:10000003; )
```

I had used single quotes (') to break the query inside the text field of username then click on **submit**.

Username: '

From the given screenshot you can see we have **got error message (in blue color)** which means the database is vulnerable to SQL injection.

For more detail on Form Based SQL injection read our previous article.

Now when attacker will execute malicious quotes in browser for testing Form Base SQL injection then the IDS of the network should also capture this content and will generate the alert.



As per our prediction from given image you can observe the snort has generated alert for Form Based sql injection when capture malicious quotes.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-00:30:22.916255  [**] [1:10000003:0] Form Based SQL Injection Detected [**]
[Priority: 0] {TCP} 192.168.1.21:37274 -> 192.168.1.20:80
```

Identify Order by SQL Injection

In order to identify number of column in database the un-trusted user may use **order by** clause which will arrange the result set in ascending or descending order of the columns used in the

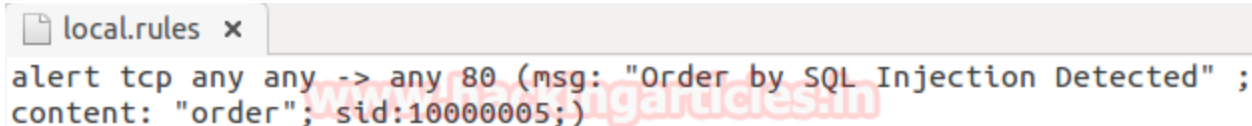
query.

Now add a rule in snort which will analyse order by SQL injection on the server when someone tries to execute SQL query in your network for unprivileged access of database. Here again **that I had applied filter for content "order" to be captured.**

```
alert tcp any any -> any 80 (msg: "Order by SQL Injection"; content: "order" ;
```

Turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



```
local.rules x
alert tcp any any -> any 80 (msg: "Order by SQL Injection Detected" ;
content: "order"; sid:10000005;)
```

Now again open the server IP in web browser and use string order by for identify column of database as shown below.

```
192.168.1.20/sqli/Less-1/?id=1' order by 1,2,3 --+
```

Now when attacker will execute malicious string in browser for testing order by SQL injection then the IDS of the network should also capture this content and will generate the alert



As per our prediction from given image you can observe the snort has generated alert for order by sql injection when capture malicious string.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-01:48:06.081322  [**] [1:10000005:0] Order By SQL Injection Detected [**]
[Priority: 0] {TCP} 192.168.1.21:37400 -> 192.168.1.20:80
```

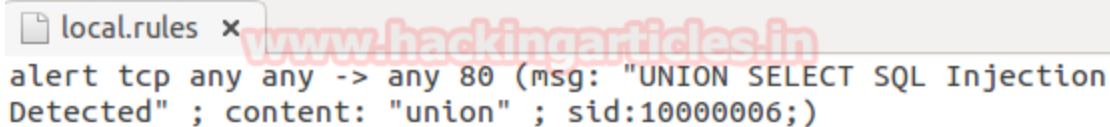
Identify Union Based SQL Injection

We all know in Error base SQL injection attacker may use the UNION operator to combine the result-set of two or more SELECT statements. Therefore add a rule in snort which will analyst Union select SQL injection on the server when someone try to execute SQL query in your network for unprivileged access of database. Here again **that I had applied filter for content "union" to be captured.**

```
alert tcp any any -> any 80 (msg: "UNION SELECT SQL Injection"; content: "union"
```

Turn on IDS mode of snort by executing given below command in terminal:

```
sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
```



```
local.rules x
alert tcp any any -> any 80 (msg: "UNION SELECT SQL Injection
Detected" ; content: "union" ; sid:10000006;)
```

Now again open the server IP in web browser and use string order by for identify column of database as shown below.

```
192.168.1.20/sqli/Less-1/?id=-1' union select 1,2,3 --+
```

Now when attacker will execute malicious string in browser for testing Union select SQL injection then the IDS of the network should also capture this content and will generate the alert.



As per our prediction from given image you can observe the snort has generated alert for union select sql injection when capture malicious string.

So when the network admin get alert from IDS on the basis of it he can take action against attacking IP, as shown in given image the malicious traffic is coming form 192.168.1.21 on port 80.

```
zed@ubuntu:/etc/snort/rules$ sudo snort -A console -q -u snort -g snort -c /etc/snort/snort.conf -i eth0
01/11-02:56:25.682946 ** [1:10000006:0] UNION SELECT SQL Injection Detected [
**] [Priority: 0] {TCP} 192.168.1.21:37438 -> 192.168.1.20:80
```

Author: Shubham Sharma is a passionate Cybersecurity Researcher, contact **LinkedIn** and **Twitter**.

◀ PREVIOUS POST

Check Meltdown Vulnerability in CPU

NEXT POST ▶

How to Configure Suricata IDS in Ubuntu

3 thoughts on “Detect SQL Injection Attack using Snort IDS”

**Miguel**

March 14, 2020 at 10:07 am

Thanks a lot.

Great info.

**Stefan**

October 23, 2020 at 5:20 am

Now days SQL injection are definitely on port 443.

How we can intercept such traffic when SNORT is in passive mode?

**sepehr**

January 18, 2021 at 12:08 am

hi im using the 127.0.0.1/ address block for apache web server and running snort on loopback interface but it wont alarm me on sql injection attacks that you've introduced in this article , but it will detect the nmap scan i run on loopback address 127.0.0.1 ! is this problem related to me using 127.0.0.1/8 ip block and not 192.168.0.0/16 for my web server or what? i would be grateful if you help me here

Comments are closed.

Join Our Training Program





Categories

Cryptography & Steganography

CTF Challenges

Cyber Forensics

Database Hacking

Footprinting

Hacking Tools

Kali Linux

Nmap

Others

Password Cracking

Penetration Testing

Pentest Lab Setup

Privilege Escalation

Red Teaming

Social Engineering Toolkit

Uncategorized

Website Hacking

Window Password Hacking

Wireless Hacking

Wireless Penetration Testing

Archives

Select Month



You may like

Best Alternative of Netcat Listener

April 3, 2024

64-bit Linux Assembly and Shellcoding

March 29, 2024